AM 255: Problem Set 2

Douglas Lanman

28 September 2006

Problem 1

Consider the first-order wave equation given on page 38 of [1].

$$u_t = u_x, \quad -\infty < x < \infty,\ 0 \leq t$$
$$u(x,0) = \sin(x), \quad -\infty < x < \infty$$

Compute the discrete difference approximation at time $T = 2\pi$ given by

$$v_j^{n+1} = (I + kD_0)v_j^n = v_j^n + \frac{\lambda}{2}\,(v_{j+1}^n - v_{j-1}^n), \tag{1}$$

where $\lambda = k/h$ is the ratio of the time step k to the space step h. Consider discrete grids of size $N = \{19, 39, 79, 159, 319\}$ and values of $\lambda = \{1/2, h\}$. Graphically compare the exact solution to the numerical solutions for $N = 19$ and tabulate the L_2-errors. Finally, estimate the order of approximation achieved for each value of λ.

My implementation of the discrete difference approximation, as defined by Equation 1, was completed using Matlab and is included as `prob1.m`. Note that `prob1.m` accepts a single input argument `kMode` which is used to toggle $\lambda = 1/2$ or $\lambda = h$.

Before presenting the results of my program, I will briefly outline the architecture of the source code. On lines 21-55 I select the values of $\{N, h, k\}$ and determine the resulting grid points $\{x, t\}$. (Note that on lines 44-47 I ensure that the last time is given by $T = 2\pi$.) Lines 56-66 implement Equation 1. Note that I have implemented the central difference operator D_0 as a stand-alone program `D0.m`. Finally, lines 67-104 create the tables and plots shown in this write-up.

Recall from Equation 2.1.3 in [1] that the general solution to the first-order wave equation is given by $u(x,t) = f(x+t)$, where $f(x)$ is the initial condition along $t = 0$. As a result, the solution at $T = 2\pi$ is given by $u(x, 2\pi) = \sin(x + 2\pi) = \sin(x)$. As specified in the problem statement, I have plotted the numerical approximation along with the analytic solution in Figure 1. (Note that the cases $\lambda = \{1/2, h\}$ are shown in Figure 1(a) and 1(b), respectively. Also recall that $\lambda = h \Rightarrow k = h^2$ from page 44 in [1].)

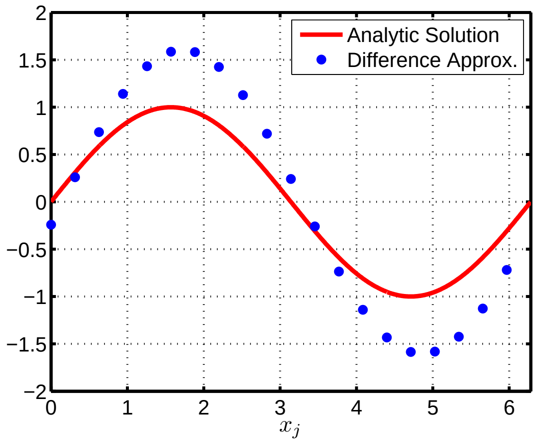
The approximation results for both $\lambda = 1/2$ and $\lambda = h$ are tabulated below.

N	L_2-error	order
19	1.121	NA
39	0.496	$h^{1.18}$
79	0.233	$h^{1.09}$
159	0.696	$h^{-1.58}$
319	1.231e15	$h^{-50.65}$

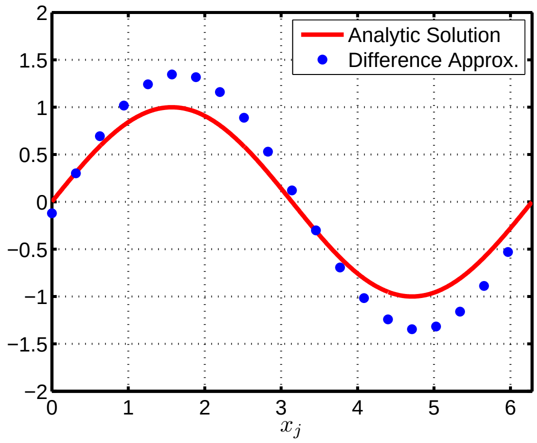
Table 1.1: $\lambda = 1/2$

N	L_2-error	order
19	0.647	NA
39	0.146	$h^{2.15}$
79	3.536e-2	$h^{2.04}$
159	8.680e-3	$h^{2.03}$
319	2.176e-3	$h^{2.00}$

Table 1.2: $\lambda = h$



(a) Results for $N = 19$ and $\lambda = 1/2$



(b) Results for $N = 19$ and $\lambda = h$

Figure 1: Comparison between difference approximations and the analytic solution.

Note that the standard definition of the discrete L_2 norm was used to evaluate the total error as

$$L_2\text{-error}(N) \triangleq \sqrt{\sum_{j=0}^{N} |u(x_j, t^n) - v_j^n|^2 h}.$$

In addition, the following definition of order of approximation was given in class.

$$\text{order} \triangleq \log_2 \left(\frac{L_2\text{-error}(N)}{L_2\text{-error}(2N)} \right)$$

In conclusion, we find that the numerical results agree with the predictions made in class and on pages 38-44 in [1]. Specifically, we find that the numerical solution is unstable for $\lambda = 1/2$, whereas it is stable for $\lambda = h$. Despite achieving stability, this solution remains undesirable as it requires too many time steps to achieve a robust estimate in practical situations.

Problem 2

Consider the discrete difference approximation to $u_t = u_x$ given by

$$v_j^{n+1} = (I + kD_0)v_j^n + \sigma kh D_+ D_- v_j^n, \text{ where } v_j^0 = f_j. \tag{2}$$

Modify this scheme such that it approximates $u_t = -u_x$. Prove that the conditions in Equations 2.1.14 and 2.1.15 from [1] are also necessary for stability in this case.

To begin our analysis, note that Equation 1 approximates the differential equation $u_t = u_x$ by taking the forward difference in time and the the central difference in space. Equation 2 incorporates an additional *artificial viscosity* term into this expression. As a result, we can approximate the differential equation $u_t = -u_x$ by changing the sign of the central difference in space as follows.

$$\boxed{v_j^{n+1} = (I - kD_0)v_j^n + \sigma kh D_+ D_- v_j^n, \text{ where } v_j^0 = f_j.} \tag{3}$$

Rearranging the terms in Equation 3, we have

$$\frac{v_j^{n+1} - v_j^n}{k} = -D_0 v_j^n + \sigma h D_+ D_- v_j^n,$$

which approximates the differential equation

$$u_t = -u_x + \sigma h u_{xx}.$$

In the limit $h \to 0$, the term in u_{xx} becomes negligible and we obtain an approximate solution to $u_t = -u_x$.

In order to find the necessary conditions for stability, we begin by making the ansatz

$$v_j^n = \frac{1}{\sqrt{2\pi}} \hat{v}^n(\omega) e^{i\omega x_j},$$

where the solution is composed of a single Fourier component. Substituting this expression into Equation 3, we obtain the following expression.

$$e^{i\omega x_j} \hat{v}^{n+1}(\omega) = (I - kD_0 + \sigma k h D_+ D_-)\, e^{i\omega x_j}\, \hat{v}^n(\omega) \tag{4}$$

Recall from [1] the following forms for the forward, backward, and central difference operators in terms of the shift operator E.

$$D_+ = (E - E^0)/h, \;\; D_- = (E^0 - E^{-1})/h, \text{ and } D_0 = (E - E^{-1})/2h \tag{5}$$

Combining these expressions, we find

$$D_+ D_- v_j^n = \frac{(E - 2E^0 + E^{-1}) v_j^n}{h^2} = \frac{v_{j+1}^n - 2v_j^n + v_{j-1}^n}{h^2}. \tag{6}$$

Applying Equations 5 and 6 to Equation 4, we obtain

$$e^{i\omega x_j} \hat{v}^{n+1}(\omega) = \left(e^{i\omega x_j} - \frac{\lambda}{2}\left(e^{i\omega x_{j+1}} - e^{i\omega x_{j-1}}\right) + \sigma\lambda\left(e^{i\omega x_{j+1}} - 2e^{i\omega x_j} + e^{i\omega x_{j-1}}\right)\right) \hat{v}^n(\omega),$$

where $\lambda = k/h$. Recall that $x_j = jh$ such that $e^{i\omega x_j} = e^{i\omega jh}$. As a result, we can factor out $e^{i\omega x_j}$ on the right-hand side of the previous expression as follows.

$$e^{i\omega x_j} \hat{v}^{n+1}(\omega) = e^{i\omega x_j} \left(1 - \frac{\lambda}{2}\left(e^{i\omega h} - e^{-i\omega h}\right) + \sigma\lambda\left(e^{i\omega h} + e^{-i\omega h} - 2\right)\right) \hat{v}^n(\omega)$$

Using the basic trigonometric identities $\sin x = (e^{ix} - e^{-ix})/2i$ and $\cos x = (e^{ix} + e^{-ix})/2$, the previous expression can be reduced to

$$\hat{v}^{n+1}(\omega) = (1 - i\lambda \sin(\omega h) + 2\sigma\lambda\,(cos(\omega h) - 1))\, \hat{v}^n(\omega).$$

Finally, we recall the half-angle formula in which $\sin^2(x/2) = (1 - \cos(x))/2$; applying this formula to the previous equation provides a closed-form expression for the *symbol* $\hat{Q}$.

$$\hat{v}^{n+1}(\omega) = \hat{Q}\hat{v}^n(\omega), \quad \hat{Q} = 1 - i\lambda \sin\xi - 4\sigma\lambda \sin^2\frac{\xi}{2}, \tag{7}$$

where $\xi = \omega h$.

Recall from page 44 in [1] that we consider a method *stable* if

$$\sup_{0 \leq t_n \leq T, \omega, k, h} |\hat{Q}^n| \leq K(T),$$

as $h, k \to 0$. As was done in the textbook, we can choose σ, k, and h such that

$$|\hat{Q}| \leq 1 \Rightarrow |\hat{Q}|^2 \leq 1. \tag{8}$$

Substituting the expression for the symbol $\hat{Q}$ from Equation 7, we derive the following expression.

$$\begin{aligned}|\hat{Q}|^2 &= \left(1 - 4\sigma\lambda \sin^2 \frac{\xi}{2}\right) + \lambda^2 \sin^2 \xi \\ &= 1 - (8\sigma\lambda - 4\lambda^2) \sin^2 \frac{\xi}{2} + (16\sigma^2 - 4)\lambda^2 \sin^4 \frac{\xi}{2}\end{aligned} \tag{9}$$

Combining Equations 8 and 9, we derive the following constraint for a stable solution.

$$(8\sigma\lambda - 4\lambda^2) \sin^2 \frac{\xi}{2} - (16\sigma^2 - 4)\lambda^2 \sin^4 \frac{\xi}{2} \geq 0 \tag{10}$$

First, consider the situation in which $2\sigma \leq 1 \Rightarrow (16\sigma^2 - 4) \leq 0$. In order to guarantee that Equation 8 is satisfied, it is sufficient for

$$8\sigma\lambda - 4\lambda^2 \geq 0.$$

$$\Rightarrow \lambda \leq 2\sigma, \text{ such that } 0 < \lambda \leq 2\sigma \leq 1, \tag{11}$$

which is precisely the stability condition specified by Equation 2.1.14 in [1]. Now, let us consider the case for which $2\sigma \geq 1 \Rightarrow (16\sigma^2 - 4) \geq 0$. In order to guarantee that Equation 8 is satisfied, it is necessary for

$$(8\sigma\lambda - 4\lambda^2) \sin^2 \frac{\xi}{2} \geq (16\sigma^2 - 4)\lambda^2 \sin^4 \frac{\xi}{2}$$

$$\Rightarrow (8\sigma\lambda - 4\lambda^2) \sin^2 \frac{\xi}{2} \geq (16\sigma^2 - 4)\lambda^2 \sin^2 \frac{\xi}{2},$$

since $\sin^4 \frac{\xi}{2}$ is bounded from above by $\sin^2 \frac{\xi}{2}$. Reducing the previous expression gives

$$8\sigma\lambda - 4\lambda^2 \geq (16\sigma^2 - 4)\lambda^2$$

$$\Rightarrow 2\sigma\lambda \leq 1, \tag{12}$$

which is precisely the stability condition specified by Equation 2.1.15 in [1]. In conclusion, we have shown that this modified scheme will be stabile if the conditions in Equations 2.1.14 and 2.1.15 from [1] are satisfied, as tabulated below.

$$\boxed{\begin{array}{l}\textbf{Condition 1: } 0 < \lambda \leq 2\sigma \leq 1 \\ \textbf{Condition 2: } 2\sigma \geq 1,\ 2\sigma\lambda \leq 1\end{array}} \tag{13}$$

(QED)

Problem 3

Choose σ in Equation 2 such that Q only uses two gridpoints. What is the stability criterion?

Let use define the symbol Q such that $v_j^{n+1} \triangleq Q v_j^n$. By inspection of Equation 2, we have

$$Q = I + kD_0 + \sigma k h D_+ D_-. \tag{14}$$

At this point, we review the *Lax-Friedrichs Method* (as presented on page 46 of [1]) for approximating $u_t = u_x$, which is given by

$$v_j^{n+1} = \frac{1}{2}(v_{j+1}^n + v_{j-1}^n) + kD_0 v_j^n. \tag{15}$$

Essentially, this approach replaces the values of v_j^n with the average of its nearest neighbors v_{j+1}^n and v_{j-1}^n. As a result, Q only uses two gridpoints to estimate v_j^{n+1}. From Equation 6, we have

$$D_+ D_- v_j^n = \frac{v_{j+1}^n - 2v_j^n + v_{j-1}^n}{h^2}.$$

Applying this expression to Equation 15, we obtain the following result.

$$\begin{aligned} v_j^{n+1} &= \frac{1}{2}(v_{j+1}^n - 2v_j^n + v_{j-1}^n) + v_j^n + kD_0 v_j^n \\ &= (I + kD_0)v_j^n + \frac{1}{2}h^2 D_+ D_- v_j^n \\ &= (I + kD_0 + \frac{1}{2}h^2 D_+ D_-)v_j^n \end{aligned}$$

By comparison to Equation 14, we have

$$\sigma k h = \frac{1}{2}h^2$$

$$\Rightarrow 2\sigma = \frac{h}{k} = \frac{1}{\lambda}. \tag{16}$$

If $\lambda \geq 1$, then $2\sigma \leq 1$ (by substitution into Equation 16). This situation contradicts Condition 1 for convergence as specified in Equation 13. As a result, we must have $\lambda \leq 1 \Rightarrow 2\sigma \geq 1$. This result satisfies Condition 2 for convergence, since $\lambda \leq 1 \Rightarrow 2\sigma\lambda \leq 1$. In conclusion, the stability criterion for Equation 15 is given as follows.

$$\boxed{\lambda = \frac{k}{h} \leq 1}$$

References

[1] Bertil Gustafsson, Heinz-Otto Kreiss, and Joseph Oliger. *Time Dependent Problems and Difference Methods*. John Wiley & Sons, 1995.

```matlab
function prob1(kMode)

% AM 255, Problem Set 2, Problem 1
%    Solves the  first-order wave equation IVP using
%    a discrete difference approximation. Results are
%    displayed graphically and tabulated for inclusion
%    in the write-up.
%
% Input:
%    kMode: Selects the mode for the time-step
%           size; kMode = {1 := k=h/2, 2 := k=h^2).
%
% Output:
%    Tables/plots required for the write-up.
%
% Douglas Lanman, Brown University, Sept. 2006

% Reset Matlab command window.
clc;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Part I: Specify discrete grid parameters.

% Specify the initial condition.
IC = @(x) sin(x);

% Define space/time grid interval(s) for evaluation.
N = [19 39 79 159 319]; % #gridpoints s.t. N+2 on [0,2*pi]
h = 2*pi./(N+1);        % resulting space steps

% Select time step (based on user input).
if ~exist('kMode','var') || kMode == 1
   k = h/2;
else
   k = h.^2;
end

% Set discrete positions/time-steps for evaluation.
% Note: All time steps will be equal, except the
%       last; it will be adjusted so that the final
%       time will be exactly 2*pi.
for i = 1:length(N)
   x{i} = h(i)*[0:N(i)];
   t{i} = [0:k(i):2*pi];
   if t{i}(end) ~= 2*pi
      t{i}(end+1) = 2*pi;
   end
end

% Initialize the numerical solution(s).
```

```
for i = 1:length(N)
   v{i} = zeros(length(t{i}),N(i)+1);
   v{i}(1,:) = IC(x{i}); % boundary values
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Part II: Evaluate difference approximation to IVP.

% Update solution sequentially (beginning with I.C.).
% Note: Uses D0.m for the central difference.
for i = 1:length(N)
   for n = 1:(length(t{i})-1)
      v{i}(n+1,:) = v{i}(n,:) + k(i)*D0(v{i}(n,:),h(i));
   end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Part III: Plot/tabulate modeling results.

% Evaluate the exact solution.
xe = linspace(0,2*pi,1000);
fe = IC(xe);

% Determine the L2-error and the approximation order.
for i = 1:length(N)
   L2_error(i) = sqrt(sum((abs(IC(x{i})-v{i}(end,:)).^2)*h(i)));
   if i > 1
      order(i) = log2(L2_error(i-1)/L2_error(i));
   end
end

% Tabulate results.
disp(' N     L2-error     order');
disp('---------------------------');
for i = 1:length(N)
   if i > 1
      fprintf('%3d   %.5g   %+2.2f\n',N(i),L2_error(i),order(i));
   else
      fprintf('%3d   %.5g\n',N(i),L2_error(i));
   end
end

% Compare approximation (N=19) to exact solution.
figure(1); clf;
plot(xe,fe,'r-','LineWidth',3);
hold on;
   plot(x{1},v{1}(end,:),'.','MarkerSize',20);
hold off;
set(gca,'LineWidth',2,'FontSize',14,'FontWeight','normal');
xlabel('$x_j$','FontName','Times',...
```

```
    'Interpreter','Latex','FontSize',16);
%title('Difference Approximation vs. Analytic Solution');
grid on; axis([0 2*pi -2 2]);
legend('Analytic Solution','Difference Approx.');
```

```
function b = D0(a,h)

% DO Central difference operator.
%    DO(A,H) evaluates the central difference of the
%    array A with grid-spacing H, as defined in:
%
%    "Time Dependent Problems and Difference Methods",
%    B. Gustafsson, H.-O. Kreiss, and J. Oliger, 1995.
%
% Douglas Lanman, Brown University, Sept. 2006

% Determine the length of the input array.
N = length(a);

% Shift array indices (modulo the array length).
fj = mod([1:N]-2,N)+1; % shift forward
bj = mod([1:N],N)+1;   % shift backward

% Evaluate the central difference.
b = (a(bj)-a(fj))/(2*h);
```